

Platform for Efficient Switching between Multiple Devices in the Intensive Care Unit

Femke De Backere¹, Thomas Vanhove¹, Emanuel Dejonghe¹, Matthias Feys¹, Tim Herinckx¹, Jeroen Vankelecom¹, Johan Decruyenaere² and Filip De Turck¹

¹ Information Technology Department (INTEC), Ghent University - iMinds, Gaston Crommenlaan 8, bus 201, 9050 Ghent, Belgium

² Department of Intensive Care, Ghent University Hospital, De Pintelaan 185, B-9000 Gent, Belgium

SUMMARY

Objectives: Handheld computers, such as tablets and smartphones, are becoming more and more accessible in the clinical care setting and in Intensive Care Units (ICUs). By making the most useful and appropriate data available on multiple devices and facilitate the switching between those devices, staff members can efficiently integrate them in their workflow, allowing for faster and more accurate decisions. This paper addresses the design of a platform for the efficient switching between multiple devices in the ICU. The key functionalities of the platform are the integration of the platform into the workflow of the medical staff and providing tailored and dynamic information at the point of care.

Methods: The platform is designed based on a 3-tier architecture with a focus on extensibility, scalability and an optimal user experience. After identification to a device using Near Field Communication (NFC), the appropriate medical information will be shown on the selected device. The visualization of the data is adapted to the type of the device. A web-centric approach was used to enable extensibility and portability.

Results: A prototype of the platform was thoroughly evaluated. The scalability, performance and user experience were evaluated. Performance tests show that the response time of the system scales linearly with the amount of data. Measurements with up to 20 devices have shown no performance loss due to the concurrent use of multiple devices.

Conclusions: The platform provides a scalable and responsive solution to enable the efficient switching between multiple devices. Due to the web-centric approach new devices can easily be integrated. The performance and scalability of the platform have been evaluated and it was shown that the response time and scalability of the platform was within an acceptable range.

MESH TERMS:

Decision Making, Computer-Assisted
Decision Support Systems, Clinical/organization & administration
Intensive Care Units
Information Systems
User-Computer Interface

CORRESPONDENCE TO:

Femke De Backere
Department of Information Technology
Internet Based Communication Networks and Services (IBCN)
Ghent University - iMinds
Gaston Crommenlaan 8 (Bus 201), B-9050 Gent, Belgium
T: +32 9 33 14938
F: +32 9 33 14899
E: femke.debackere@intec.UGent.be

1 INTRODUCTION

Handheld computers, such as tablets and smartphones, are becoming more and more popular, even in the clinical care setting [1][2][3]. Moreover, with the increasing memory capabilities, processing power and connectivity these devices can offer a portable platform for patient management in the Intensive Care Unit (ICU) [4]. Furthermore, in a computerized ICU, a computer is located next to every bed. Each department also has a unit PC and physicians usually have a personal desktop and smartphone. Moreover, the number of devices on an ICU is steadily increasing in recent years [5]. Therefore, there is a need for an efficient switching mechanism between the different devices, used in the ICU, to ensure quality of care.

These devices have the capabilities and potential to be integrated within existing clinical decision support systems (CDSS). CDSS are computer-driven technology solutions, developed to provide support to physicians, nurses and patients using medical knowledge and patient-specific information. Thus, these systems will not replace the medical staff, but will merely give advice and guidance. This way, they are able to take all relevant data and information into account. By filtering the information in an intelligent manner and presenting it to the medical staff at the appropriate moment and in an intelligent way, these systems can improve health care [6]. CDSS can be used in every aspect of the care process, from preventive care and diagnosis to monitoring and follow up. Studies have already shown that these systems improve quality, safety and effectiveness of medical decisions, resulting in improved patient care, higher performance of the medical staff and more effective clinical services [7]. Nevertheless, the uptake of CDSS is rather low and this is due to a number of factors [8].

First, one of the main problems in the use of CDSS is the integration of applications into the current workflow of the medical staff [9]. Kawamoto et al. [10] concluded that CDSS are more successful when integrated into the work process of the medical staff. This also means the integration with existing information systems of the hospital [11].

Second, the devices used in the ICU are not optimal embedded within CDSS. Sharing information at the right time and place has a large influence on the use of these systems and on the performance of the medical staff, moreover it is time-saving [12].

A third problem is representing all the relevant information of a specific patient. In the ICU, up to 200,000 parameters are collected for each patient on a daily basis [13][14]. These parameters are mainly originating from examinations and from monitoring data. Visualizing this data in an optimal way and selecting only the most relevant information is a challenging task [15].

Due to problems and the necessity to ensure continuity of healthcare services, improving patient quality of life and rationalizing healthcare costs, new pervasive healthcare systems [16] are being explored [17][18]. The research on clinical decision support has evolved over 50 years [19] resulting in new approaches such as pervasive and ubiquitous healthcare [20][21].

As the medical staff in a clinical setting has very diverse tasks and the work is highly fragmented [22][23]. On average, they do not spend more than 5 minutes on a specific task

and in many cases only spend 1.5 minutes on a specific activity before switching to another task. This means that personnel has to be highly adaptive and should be able to cope with an ever changing environment and continually adjust their activities [24]. In an intensive care setting, there is a wide variety of systems that are integrated in the workflow of the doctors and the nurses. This means that during an activity they are taken into account the readings and/or measurement from these systems into account to assess the situation of the patient or they have to interact with different (software) systems to obtain the correct information about a specific patient [25]. As the staff only has limited time to spend on certain tasks or activities, accessing the correct infrastructure and tools can create a big overhead for the staff. The study of Koch et al. [26] indicated that using integrated displays, where all important information is contained in one screen, could be an advantage, if bidirectional communication between different devices is implemented. Also, event recognition and treatment efficiency can be improved when using a second display [27].

A better integration of the current infrastructure and handheld devices, such as tablets and smartphones, can improve this situation. Therefore, there is a need for a platform, which is capable of switching between the different devices, used in the ICU. However, there are still obstacles concerning the efficient switching between devices and that should be taken into account when developing such a platform. First, the user friendliness should be of paramount importance. Introducing a new tool into such a complex setting as an ICU, should improve the quality of care and the workflow of the medical staff. It should support the staff in their current activities. Second, the speed of the switching mechanism is also important, as doctors and nurses only spend 5 minutes on average on a task, they do not want to wait for a few minutes while transferring the data from one device to another. Finally, the switching mechanism should be carried out in such a manner, that it suggest when the situation is right to switch to another device and automatically detect other devices in the vicinity. Keeping user friendliness in mind, users should get the suggestion to switch and should not be switched to another device automatically.

The purpose of this paper is twofold. On the one hand, the design and implementation of a platform is presented, enabling doctors and other members from the medical staff to switch between multiple devices. This platform is also capable of detecting which content is suitable to be displayed on which device, e.g., text can be shown on all devices, while high-resolution images are less suitable to be displayed on smartphones. On the other hand, the performance of the platform is evaluated, to give valuable insights in the scalability, responsiveness and user experience.

The remainder of the article is structured as follows. Section 2 details the objectives of our platform, whereas Section 3 is devoted to the methodological approach. Section 4 deals with the evaluation results. Finally, the main contributions are discussed and the main conclusions of this research are highlighted in Sections 5 and 6.

2 OBJECTIVES

The aim of this research is to design a platform that allows for the efficient switching between devices in an Intensive Care setting. This platform should offer following features:

- Integration in the workflow and at the point of care: to optimize the care and minimize loss of time and costs, the visualization of the data by the platform should be integrated into the workflow of the doctors and the medical staff. Moreover, the

visualization should also be possible at the point of care, while examining the patients and not only in the office of the physician.

- Tailored information: The information should be adapted dynamically to the capabilities of the devices, e.g., high quality images, such as X-ray pictures, should not be shown on a smartphone as the transfer of data would create an unacceptable delay and it would not be easy to interpret this kind of images on a small screen.
- Dynamic information: When new information becomes available, the device should be able to immediately visualize the new content.
- Displaying the appropriate medical information on the device: Based on the role and the preferences of the end-user and the properties of the device, which are provided to the platform by means of a database, the platform is capable of selecting and visualizing the information in a user friendly manner. By enabling the medical staff to enter their personal preferences, we make sure that it possible to deviate from the settings made by the platform and we ensure a user-friendly experience. For example, a cardiologist should see information concerning the heart instead of seeing kidney data first.

Besides these functional requirements, the platform should also fulfill the following non-functional requirements:

- The platform should be generic and it should be possible to plug-in new devices at any moment.
- As the number of devices in the ICU is increasing at a steady pace, the platform should be scalable and able to cope with a large number of clients.
- The platform performance should be such that the loading times are in an acceptable range.

The original research contribution of the paper is the design of a platform for the efficient switching between devices in an Intensive Care setting, taking into account the above four functional requirements and the three non-functional requirements. The design of the platform is outlined in the paper and obtained performance results are presented, together with a discussion section. The platform can also be used outside the intensive care setting, for instance in ambulatory settings.

3 METHODS

The platform offers an environment, in which the efficient switching between devices is facilitated. Section 3.1 details the general concept of the platform's architecture, whereas Section 3.2 describes the platform components and their interactions, by focusing on the envisioned scenarios. Section 3.3 discusses the use of Near Field Communication as a localization and switching standard between the devices. In Section 3.4, the components of the platform and their interactions are described. Further implementation details, concerning the three-tiered architecture are given in Sections 3.4.1, 3.4.2 and 3.4.3. Section 3.4.4 handles the implementation details of the NFC communication. Finally, Section 3.5 details the security and confidentiality techniques used within the platform.

3.1 GENERAL CONCEPT

[Figure 1](#) illustrates the general concept of the platform. As can be seen in this figure, data from a various range of sources is gathered in the Intensive Care Information System (ICIS). This involves data from clinical observations, prescription information, monitoring parameters, lab results as well as administrative data. Furthermore, information regarding the personal preferences from doctors is stored in the Staff Preferences database and general knowledge is kept in the Knowledge database, for example the capabilities of every type of device. The information from these three databases is used for filtering and selecting the requested information. Based on the capabilities of device and the preferences of the user, the information can be filtered in an additional step, if necessary, and is sent to the device.

3.2 SCENARIO

From the general concept, as discussed in the previous section, the following scenario can be envisioned.

- 1) The doctor is on his way for his round in the ICU ward and decides that he already wants to check the last measurements of patient X. He takes his smartphone and gets a concise overview, in the form of a table, of the patient's status.
- 2) As the doctor arrives at the ICU ward, he wants to visualize the measurements onto the bedside desktop PC. Therefore, he swipes his personal tag on the reader, attached to the device.
- 3) Immediately, the patient's data that the doctor was viewing on the smartphone is shown on the bedside PC. As this screen has a larger size, the measurements are shown as graphs, where possible.
- 4) Meanwhile, the nurse at the unit desktop PC is entering additional information about patient X. Straightaway, all devices, currently visualizing data about patient X, will be updated, ensuring an up to date view on patient X.
- 5) After a while, the doctor moves to the bed of patient Y and on the bedside PC he visualizes this patient's data. With this action, the smartphone application, still visualizing the data of patient X, will refresh automatically and instantly show the data of patient Y.

The visualization of this scenario is shown in [Figure 2](#). Next to the floor plan, some examples are given of which information is displayed on the screen of the involved devices.

The scenario, described in the previous paragraph, details the general concept of the proposed platform. However, the platform makes it possible to switch between a wide range of different devices that could be used in the ICU: smartphones, tablets, desktop computers at the nurse's station or in the doctor's office, bedside pcs and smart TVs. In fact, all devices, which are capable of visualizing web pages can be used with the platform. Different uses cases for switching are:

- Switching between devices, from a device with a small screen to one with a bigger screen, because the medical staff wants to have a more detailed overview of certain variables. This can be done by means of a graph instead of a table or a listing of the variables from the last hour.
- Switching from a screen that can be seen by visitors of patients, during visiting hours, to a more personal device. This way, patient confidentiality can be taken into account.

- Switching from a device, residing next to the patients bed, to a more personal device, because the doctor is continuing his/her round.

3.3 NEAR FIELD COMMUNICATION

To implement the tags and readers, as mentioned in the previous section, we assume doctors and the medical staff will use Near Field Communication (NFC)[28]. NFC is a new set of standards that enables smartphones and other devices, with similar capabilities, to establish radio communication. This connection is set up by bringing the devices in close proximity to each other (usually only a few centimeters), or by touching each other. Not only communication between two NFC enabled devices is possible, but also the communication between an NFC reader and an unpowered NFC chip, which is often called a tag.

NFC is compatible with current existing Radio Frequency Identification (RFID) structures, tags and smart cards [29][30]. There also is no technical barrier to use NFC, as the concept is straightforward. The user just has to bring the two devices in their range to start communication. As the communication range is short, it is easy to distinguish multiple devices residing in each other's neighborhood. This also means that there is little change that there will be security issues, if no other device is in the vicinity, there will be no communication [31].

3.4 PLATFORM COMPONENTS AND INTERACTIONS

The platform is implemented by using Java EE 6 (Java Enterprise Edition 6), which defines a standard for developing and implementing multi-tier applications, based on standardized modular components. The Java EE framework offers a complete set of services to these components and details concerning middleware activities are handled automatically, without complex programming. A multi-tier, distributed application model is used by this platform. Based on functionality, the application is split up into different components, which can be installed on different machines, depending on the tier they belong to. Most Java EE enterprise applications can be split into 3 tiers:

- Entities are contained in the *Persistence Tier*. The Java Persistence API is used to implement entities and to persist them into a table in a relational database.
- Enterprise Java Beans (EJB) are defined in the *Business Tier*. These beans are responsible for adding logic to the application. The Java EE framework ensures that these EJBs offer scalability by means of resource pooling. There are two different types of EJBs. Tasks of clients are performed by session beans. Based on the requirements, session beans can be stateless, stateful or a singleton. Message-driven beans are used when the application has to process asynchronous messages. In the Business Tier, Web Services can be defined, which can call upon external services.
- Java Server Pages (JSP) and servlets are stored in the *Web Tier*. JSP and servlets can be used to visualize dynamic web content and make it possible to enforce a separation between the representation of data and the business logic.

The 3-tier architecture of the platform, as depicted in [Figure 3](#), is based on a web centric approach and addresses the functional needs as mentioned in Section 2. These choices ensure that the platform is flexible and portable. Also, by choosing a web centric approach, all

devices with a web browser are able to plug into the system. The implementation code of the Proof of Concept is made publically available on GitHub through the following URL: <https://github.ugent.be/fddbacke/DeviceSwitching.git>.

3.4.1 PERSISTENCE TIER

The Persistence Tier contains all the entities, representing the tables in the database. In fact, entities are Plain Old Java Object (POJO), extended with annotations that can indicate for example, an ID or multiplicities, such as a many-to-many relationship. The most important entities in the platform are:

- *DeviceType*: information about the specifications of a device type, e.g., the resolution of the screen.
- *Staff*: knowledge about staff members' preferences and limitations, concerning the data that a certain staff member can consult.
- *Variable*: identifier to determine which type of data is stored, for example, body temperature, blood pressure, heartbeat per minute.
- *Patient*: detailed information about the patient, such as name, episode number and unique national number.
 - *PatientVariable*: actual data about a specific *Variable*, linked to the *Patient*. For example: Patient x has a body temperature of 37.2 degrees.

3.4.2 BUSINESS TIER

In the Business Tier, the Facade design pattern is applied [32]. This implies that all communication between the Web Tier and Business Tier will pass through a specific bean, the *ManagementBean*. This bean provides only high-level business methods in order to have a safe and simple interface. Furthermore, the methods in the *ManagementBean* can be split in two types. First, there are methods that are related to the state of the application for a current user. Second, there are also methods for retrieving and changing information related to the patients. Since this defines a clear distinction, the *ManagementBean* is connected to two different internal components.

The first component, the *StateManager*, keeps track of the current run-time state of the application. This class is implemented as a Singleton. A Singleton session bean is instantiated only once for each application and will exist for the whole lifecycle of that application. The Singleton session bean was chosen since this bean has to keep the information about the global state of the application. Information about the current number of devices and the users logged in on these devices is not stored in the database, but is all stored inside the *StateManager*.

Run-time state information is not stored in the database as this would result in an extra delay. The disadvantage of this decision is that information about which patient the user was viewing and which devices were in use by the user, will be lost when the server has to reboot.

The second component, the *DataBean*, is used to communicate with the Persistence Tier. This way, the application can take care of the data related to patients and staff members. Because all global state-information is saved in the *StateManager*, the *ManageBean* can be implemented as a stateless session bean. This design choice makes the application more scalable, since there can be more instances of this implementation available at the same time.

3.4.3 WEB TIER

The Web Tier consists of both code running on the device (the client) and code running on the server.

The Asynchronous Javascript and XML (AJAX) design principle is used to create a fast and responsive system [33]. However, Javascript Object Notation (JSON) was used instead of XML to enable easier processing in the client [34]. A static HTML page (HyperText Markup Language) is downloaded to the client and the content of this page is changed dynamically. Communication with the server happens in the background, without any user intervention.

The server-side code consists of servlets running in the Java EE Web Container. These will handle the requests from the client and forward them to the Business Tier. Another function of the servlets is to convert the *raw* data from the Business Tier into a representation that the client can process. This conversion depends on the type of device. For example, a graph will be generated from the raw data for tablets and laptops, while smartphones will receive a textual representation.

The data is presented to the user in the form of `Blocks`. Each `Block` contains a specific *Variable*, for example blood pressure or body temperature. When a user adds new information to a `Block`, this information is sent to the Business Tier and all devices displaying this information will receive the updated content.

Clients also poll the server at fixed intervals for new content. To lower the load on the Business Tier, a *ChangeTracker* object is introduced. The goal of this object is to keep track of which devices need to be updated with new content. When a client checks whether there is new content, the request will only be passed on to the Business Tier, if the *ChangeTracker* indicates that new content is available. This reduces the load on the server.

3.4.4 NFC COMMUNICATION IMPLEMENTATION

In order to establish a connection between the NFC infrastructure, used for the platform and the platform itself, the IOTOPE¹ library is used. By developing a small client for the device, equipped with an NFC reader, the platform can be notified, when a certain device is in the vicinity of an NFC tag. Each NFC reader can decode tags. Whenever this situation occurs, the client will perform a post to the NFC login server by transferring all the necessary data. This is in fact standard IOTOPE functionality. This login server will send the request to the login server of our platform and thus establish the connection.

The NFC communication within this platform is implemented in such a way that there is support for two different setups. In the first setup, each user has a tag and a reader is connected to each device. In the second setup, each device has a tag and a user can log in by scanning the tag with his/her personal smartphone.

The implementation of this small client is also available on <https://github.ugent.be/fddbacke/DeviceSwitching.git>.

3.5 SECURITY AND CONFIDENTIALITY

¹ <https://github.com/alexvanboxel/iotope-node/downloads>

In a clinical setting, security and privacy play an important role, as it is not the intention that everybody can gain access to the information used within the platform. Therefore, some security and privacy measurements are taking within the platform. The web application of the platform, running on the handheld device, is facilitated by means of a web page. Thus, it is possible to use existing, standard security methods for website. A secure Transport Layer Security² (TLS) connection is used. For the web application, this means that the Hyper Text Transfer Protocol (Secure) (HTTPS) is utilized. To prevent not-authorized users, accessing the application, a login mechanism is implemented. This way, only personnel of the ICU can gain access to the web application. Data can only be fetched from the databases of the ICU, if the correct WiFi hotspot is used. No information is cached in the handheld device. When the medical staff moves to another patient, the data is automatically forgotten. When a nurse or doctor forgets to leave the handheld device at the ICU, the web application will notice this when they leave the hospital, by means of GPS tracking, which is only possible outside buildings, and the web application will be closed.

4 RESULTS

Measurements were performed to evaluate the performance of the platform. The results of these tests give valuable information about the scalability, responsiveness and the user experience.

4.1 EVALUATION APPROACH

Time measurements were performed to benchmark the platform. Therefore, timestamps were collected both at the client-side and server-side. This enables the calculation of the response time under different circumstances and the determination of the most time consuming parts of the application. Special care has been taken to assure that a possible mismatch between the time on the client and the time on the server is excluded from the measurement results.

All evaluation tests were repeated 30 times. These results were then averaged to exclude statistical fluctuations. Special code was included in the client to facilitate these measurements. This code simulates user interaction and collects timestamps belonging to the performed action. Reaction time of the NFC reader and the time to render the Document Object Model (DOM) in the client browser are excluded from the measurements.

Response time will always be used as benchmark. This is defined as the time between the selection of the patient and the time when the DOM with the patient's information is updated in the client.

4.2 EVALUATION SETUP

All the tests were performed with the same server: a laptop with an Intel Core i5-2410M CPU with 6 GB RAM and a SSD running Microsoft Windows 7. Glassfish Version 3.1.2 was used as the Application Server. The client used in the tests, unless stated otherwise, is a laptop with an Intel Core i5-3210M running Ubuntu 12.04. Firefox 17.0 was used as web browser. For the communication between the devices a wireless 802.11 b/g router with 100Mbit/s Ethernet connection was used. Unless stated otherwise, the devices were connected to the WiFi network.

² <http://tools.ietf.org/html/rfc5246>

A Samsung Galaxy SII Smartphone, running Android 4.0 and a Samsung Galaxy Tab 10.1 tablet, running Android 3.1 were used as mobile devices. The tests were performed using their standard web browsers.

4.3 EVALUATION RESULTS

The first evaluation analyzes the response time in function of the number of blocks that are sent to the client. This test was performed with a wired connection between server and client. As can be seen in [Figure 4](#), the relation between the response time and the number of blocks shown in the client is linear. Presenting the data as graphs takes more time than presenting the data in a tabular form. The response time increases also faster when graphs are generated. This was expected since the number of blocks equals the number of graphs to be created. Also, the time for generating three blocks, is almost equal for both representations. This is because the first three blocks never contain any graphs.

To gain more insights in the response time, the different parts were analyzed. Three different parts were identified:

- DOM-modification time at the client-side
- The communication delay between server and client. This delay consists of both the network delay and the JSON-parsing at the client-side.
- The generation at the server-side, this can further be split in:
 - Switch and select patient at the server-side
 - Business logic without database interaction
 - Interaction with the database
 - Generating HTML

The result of this analysis with the data represented as graphs, or as tables can be found in [Figure 5](#) and [Figure 6](#) respectively. The average and standard deviation are shown in [Table 1](#). It can be concluded that for the rendering of tables the most time-consuming parts are the DOM modification at the client-side and the database interaction at the server-side. For the rendering of graphs the HTML-generation at the server-side takes a considerable amount of time. This is caused by the fact that the generation of the graphs is done as part of the generation of the HTML code. At the server-side, the database interaction always takes a considerable part of the total response time.

The average response times were measured when different devices were connected with each device constantly sending requests to the server. With up to 20 devices connected, no statistically significant differences in response time were measured, implying that the system scales well.

Finally, the response time was measured on different types of devices to evaluate the user experience. In normal operation, the system will show a different representation of the data depending on the type of device. For these measurements, the same data was sent to each device to enable comparison of the performance of the different devices. This data consists of 25 blocks, each containing 100 values and these are represented as a graph and as a table. The results are shown in [Table 2](#). On all devices the representation with graphs consistently takes longer. The smartphone also performs better than the tablet for both the representation with graphs and tables.

5 DISCUSSION

An accurate analysis of the evaluation results, obtained as described in the previous section, indicates that the platform performs as desired. The response time for generating tables, as shown in [Figure 6](#), shows that most of the time goes to client DOM modification (57%) and database interaction (31%). For generating graphs, as shown in [Figure 5](#), 65% of the response time is needed for HTML generation. Client DOM modification and database interaction take respectively 12% and 16% of the response time.

Measurements with up to 20 devices have shown no performance loss due to the concurrent use of multiple devices, proving that the application is scalable. The ICU of Ghent University³ is one of the largest ICUs in Belgium, which holds 56 beds. The department consists of 5 different units (cardiac, burn unit, surgical, pediatric and internal) and these are located in different locations. Each unit in itself is again divided in several smaller units where each ICU bed has a bedside PC. As the platform can be distributed in such a way that each small unit has its own server, it can be ensured that the platform keeps running smoothly. Moreover, in the experiments where 25 medical parameters were measured and 100 values per parameter were stored, the system was able to consistently respond in less than 1 second when the data is presented in tabular form.

The difference in response time between the different types of devices is less than a factor 3, as can be observed in [Table 2](#). This difference can be partly compensated by adapting the representation of the data to the device type, which guarantees a consistent user experience on all devices. Furthermore, the performance difference between the tablet and smartphone, used in this experimental set-up, can most probably be explained by the newer Android version and the faster CPU of the smartphone.

When the response time as a function of the number of the data points is measured, as shown in [Figure 4](#) and [Table 1](#), a linear relationship is achieved as expected. The HTML generation takes 50 times as much time than presenting the data in a tabular form. The increase in response time as a function of the number of data points is smaller with graphs, since the number of graphs that needs to be created is the same.

The main advantages of using the new NFC technology, instead of the more known RFID technology are: (i) the capability of bi-directional communication, this way of communication, instead of single mode communication as with RFID, allows for more flexibility as the tags are able to communicate directly with each other, (ii) the ability to emulate contactless smart cards, which advances the interoperability of NFC as there is no need for an NFC tag or RFID card and information, stored on the NFC device, is used for communication, however, making the system secure is a great challenge [35](iii) peer-to-peer connections can be established, with this mode it is possible to exchange data at link-level [36][37], and (iv) the speed of the connection establishment is negligible, NFC connections are typically set up in less than 100 milliseconds as the connection between 2 devices is created automatically[38].

The platform performance can be further improved by reducing the number of requests from the client to the server: this can be realized by changing the polling architecture (from client to server) to a push-based architecture (from server to client). In this way, when the server discovers a change to the data of a viewer, it will push these changes to the client.

³ <http://www.icu.be/eng/>

The following technical considerations and challenges should be taken into account for the platform. First, as the platform is used within a clinical setting, privacy and security are very important. Therefore, we also plan to carefully consider the privacy and security requirements in the extended version of the platforms. Second, Javascript was chosen to create a fast and responsive system, and the execution of the code is done at client side, which limits the necessary processing power in the back-end. However, this also means that the CPU of the end user's device is used, which can have an impact on the battery consumption of mobile client. An adaptive approach to balance components between client server is currently being considered. Another limitation of Javascript is that different layout engines will render the code in a different manner, which may result in inconsistencies in terms of functionality and interface. Proper front-end development tools and extensive automated software testing will allow circumventing these incompatibility concerns. Third, by using AJAX, network latency can impact the responsiveness of the platform. Lightweight alternatives are currently being studied.

The proposed platform can be integrated within existing CDSS that are already deployed in the intensive care unit, as previous research indicates that stand-alone CDSS are not enabled to be executed on multiple computing platforms [39]. This can be done by generating the entities, based on the relational databases of the ICU. As indicated in [Figure 1](#), several databases are integrated within our platform. If these databases are replaced with those used in CDSS and the queries are adjusted accordingly, the platform can be fully operational in the ICU, integrated with the other tools and systems. EHR (Electronic Health Record) applications are considered to be an important part of CDSS, hence integrating the proposed platform with existing EHR applications can be realized in a similar way: generating the entities in the platform based on the database tables in the EHR application. This approach was taken for the integration of the system with the EHR application in the Intensive Care department of Ghent University hospital.

6 CONCLUSIONS

In this paper, a platform to access data through multiple devices is described. Based on the functional and non-functional requirements, a 3-tier architecture was designed and implemented. Due to the web-centric approach, the platform is portable, scalable and extensible. Extensive timing measurements were performed to investigate the response time, the scalability and user experience of the designed platform. The results of these evaluations show that the response time of the platform scales linearly with the amount of data. The response time for generating tables is always less than the response time for generating graphs. The platform, presented in this paper, facilitates the use of multiple devices in an ICU setting, which is integrated in the workflow of the doctors and the medical staff at the point of care. Future research will focus on replacing the polling architecture into a push mechanism and on the implementation of several caching strategies.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

- [1] Lapinsky, S.. Mobile computing in critical care. *Journal of Critical Care* 2007;22(1):41–44.
- [2] Berger, E.. The iPad: Gadget or medical godsend? *Annals of Emergency Medicine* 2010;56(1):A21–A22.
- [3] Kubben, P.. Neurosurgical apps for iPhone, iPod touch, iPad and Android. *Surgical Neurology International* 2010;1(1):89+.
- [4] Lapinsky, P., Weshler, J., Mehta, S., Varkul, M., Hallett, D., Stewart, T.. Handheld computers in Critical Care. *Journal of Critical Care* 2001;5:227+.
- [5] Colpaert, K., Vanbelleghem, S., Danneels, C., Benoit, D., Steurbaut, K., Van Hoecke, S., et al. Has information technology finally been adopted in Flemish Intensive Care Units? *BMC Medical Informatics and Decision Making* 2010;10(62).
- [6] Osheroff, J., Teich, J., Middleton, B., Steen, E., Wright, A., Detmer, D.. A roadmap for National Action on Clinical Decision Support. *JAMIA* 2007;14(2):141–145.
- [7] Garg, A., Adhikari, N., McDonald, H., Rosas-Arellano, P., Devereaux, P., Beyene, J., et al. Effects of Computerized Clinical Decision Support Systems on Practitioner Performance and Patient Outcomes. *JAMA* 2005;293(10):1223–1238.
- [8] Moxey, A., Robertson, J., Newby, D., Hains, I., Williamson, M., Pearson, S.. Computerized Clinical Decision Support for Prescribing: Provision does not guarantee Uptake. *JAMIA* 2010;17(1):25–33.
- [9] Orwat, C., Graefe, A., Faulwasser, T.. Towards Pervasive Computing in Health Care - A Literature Review. *Social Science Research Network Working Paper Series* 2008;8(26).
- [10] Kawamoto, K., Lobach, D.. Clinical Decision Support provided within Physician Order Entry Systems: A Systematic Review of Features Effective for changing Clinician Behavior. In: *AMIA Annl Sym Proc.* 2003, p. 361–365.
- [11] Decruyenaere, J., DeTurck, F., Vanhastel, S., Vandermeulen, F., Demeester, P., De Moor, G.. On the design of a generic and scalable multilayer software architecture for data flow management in the Intensive Care Unit. *Methods of information in medicine* 2003;42(1):79–88.
- [12] Kawamoto, K., Houlihan, C., Balas, A., Lobach, D.. Improving Clinical Practice using Clinical Decision Support Systems: A Systematic Review of Trials to identify Features Critical to Success. *BMJ (Clinical research ed)* 2005;330(7494):765+.
- [13] Van Hoecke, S., Decruyenaere, J., Danneels, C., Taveirne, K., Colpaert, K., Hoste, E., et al. Service-oriented subscription management of medical decision data in the Intensive Care Unit. *Methods of Information in Medicine* 2008;47(4):364–380.
- [14] Herasevich, V., Pickering, B., Dong, Y., Peters, S., Gajic, O.. Informatics Infrastructure for Syndrome Surveillance, Decision Support, Reporting, and Modeling of Critical Illness. *Mayo Clinic Proc* 2010;85(3):247–254.
- [15] Sittig, D., Wright, A., Osheroff, J., Middleton, B., Teich, J., Ash, J., et al. Grand Challenges in Clinical Decision Support. *J Biomed Inform* 2008;41(2):387–392.
- [16] Bardram, J.. Pervasive Healthcare as a Scientific Discipline. *Methods of Information in Medicine* 2008;47(3):178–185.
- [17] Triantafyllidis, A., Koutkias, K., Chouvarda, I., Maglaveras, N.. An Open and Reconfigurable Wireless Sensor Network for Pervasive Health Monitoring. *Methods of Information in Medicine* 2008;47(3):229–234.
- [18] Blobel, B.. Architectural Approach to eHealth for Enabling Paradigm Changes in Health. *Methods of Information in Medicine* 2010;49(2):123–134.
- [19] Mitchell, J., Gerdin, U., Lindberg, D., Lovis, C., Martin-Sanchez, F., Miller, R., et al. 50 years of informatics research on decision support: What's next. *Methods of Information in Medicine* 2011;50(6):525.
- [20] Peek, N., Swift, S.. Intelligent data analysis for knowledge discovery, patient monitoring and quality assessment. *Methods of Information in Medicine* 2012;51(4):318.
- [21] Surján, G., et al. How to use health informatics to manage the information overflow created by itself? *Methods Inf Med* 2013;52:97–98.
- [22] Tentori, M., Favela, J.. Activity-aware computing for healthcare. *IEEE Pervasive Computing* 2008;7(2):51–57.
- [23] Tentori, M., Hayes, G.R., Reddy, M.. Pervasive computing for hospital, chronic, and preventive care. *Foundations and Trends in Human- Computer Interaction* 2012;5(1):1–95.
- [24] Bardram, J.E., Bossen, C.. Mobility work: The spatial dimension of collaboration at a hospital. *Computer Supported Cooperative Work (CSCW)* 2005;14(2):131–160.
- [25] Bardram, E.. The trouble with login: on usability and computer security in ubiquitous computing. *Personal and Ubiquitous Computing* 2005;9(6):357–367.

- [26] Koch, S.H., Weir, C., Westenskow, D., Gondan, M., Agutter, J., Haar, M., et al. Evaluation of the effect of information integration in displays for icu nurses on situation awareness and task completion time: A prospective randomized controlled study. *International Journal of Medical Informatics* 2013;82(8):665–675.
- [27] Effken, J.A., Loeb, R.G., Kang, Y., Lin, Z.C.. Clinical information displays to improve ICU outcomes. *International journal of Medical Informatics* 2008;77(11):765–777.
- [28] Falke, O., Rukzio, E., Dietz, U., Holleis, P., Schmidt, A.. *Mobile Services for Near Field Communication*. Tech. Rep. LMU-MI-2007-1; Vodafone Group Research and Development, Munich, Embedded Interaction Research Group, University of Munic, Computing Department, Lancaster University, UK, Fraunhofer IAIS, Sankt Augustin and b-it, University of Bonn; 2007.
- [29] Ok, K., Coskun, V., Aydin, M.N., Ozdenizci, B.. Current benefits and future directions of NFC services. In: *Education and Management Technology (ICEMT), 2010 International Conference on*. IEEE; 2010, p. 334–338.
- [30] NFC-Forum, Available: <http://www.nfc-forum.org>.
- [31] Csapodi, M., Nagy, A.. New applications for nfc devices. In: *Mobile and Wireless Communications Summit, 2007. 16th IST*. IEEE; 2007, p. 1–5.
- [32] Gamma, E., Helm, R., Johnson, R., Vlissides, J.. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley; 1994.
- [33] Paulson, L.. Building Rich Web Applications with AJAX. *Computer* 2005;38(10):14–17.
- [34] Wang, G.. Improving Data Transmission in Web Applications via the Translation between XML and JSON. In: *2011 Third International Conference on Communications and Mobile Computing*. IEEE; 2011, p. 182–185.
- [35] Roland, M. "Software card emulation in NFC-enabled mobile phones: great advantage or security nightmare." *Fourth International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use*. 2012.
- [36] Madlmayr, G., Langer, J., Kantner, C. Scharinger, J.. "NFC devices: Security and privacy." In: *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*. IEEE, 2008. p. 642–647.
- [37] Ok, K., Coskun, V., Aydin, M. N. Ozdenizci, B "Current benefits and future directions of NFC services." In: *Education and Management Technology (ICEMT), 2010 International Conference on*. IEEE, 2010. p. 334–338.
- [38] Want, R.. *Near Field Communication*. IEEE Pervasive Computing 2011;10(3):4–7.
- [39] Farion, K., Michalowski, W., Wilk, S., OSullivan, D., Rubin, S., Weiss, D.. Clinical decision support system for point of care use. *Meth- ods Inf Med* 2009;48:381–390.

FIGURES

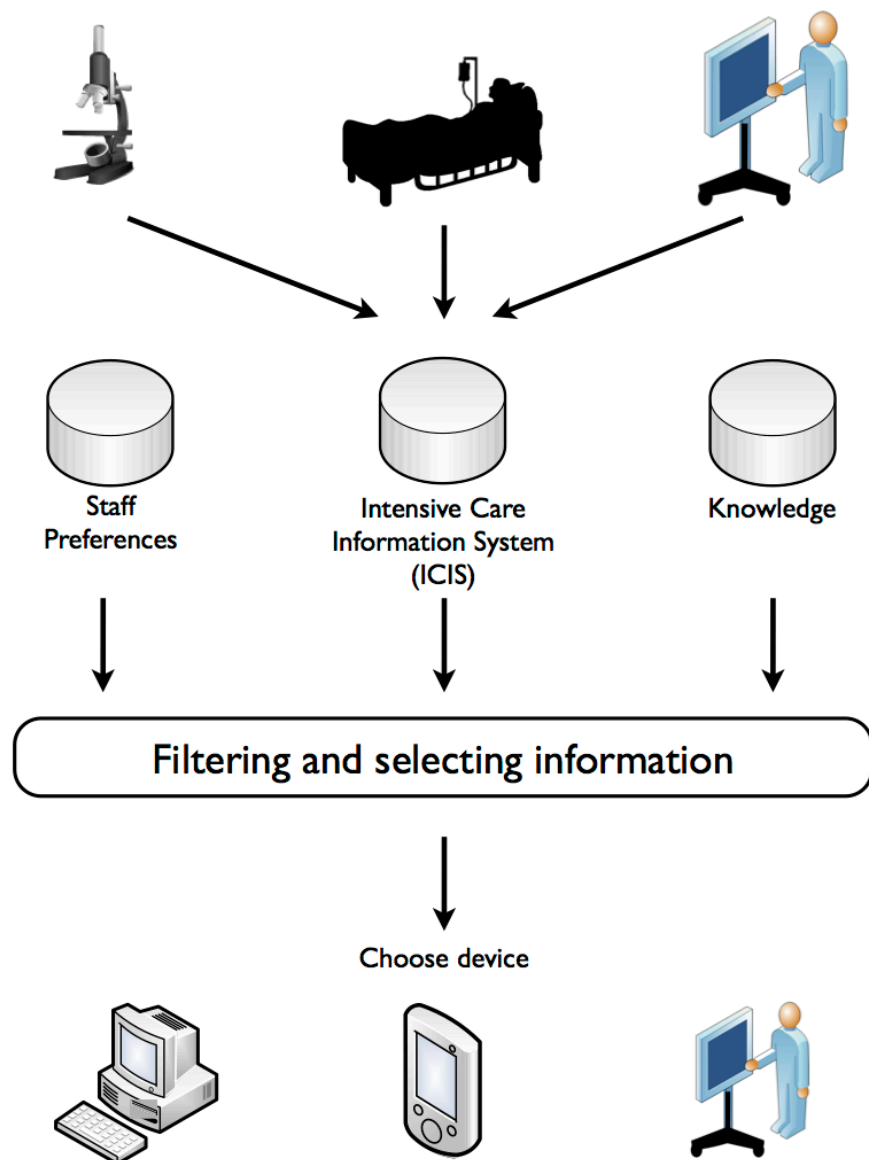


FIGURE 1 - GENERAL CONCEPT OF THE PLATFORM. DATA FROM A WIDE RANGE OF SOURCES IS GATHERED IN THE INTENSIVE CARE INFORMATION SYSTEM (ICIS). PERSONAL PREFERENCES OF THE STAFF ARE STORED IN THE STAFF PREFERENCES DATABASE AND GENERAL KNOWLEDGE IS KEPT IN THE KNOWLEDGE DATABASE. INFORMATION OF THESE DATA SOURCES IS USED TO SELECT AND FILTER THE DATA INTELLIGENTLY. BASED ON THE TYPE OF THE DEVICE AND THE PREFERENCES OF THE USER THE INFORMATION IS VISUALIZED ON A SPECIFIC DEVICE.

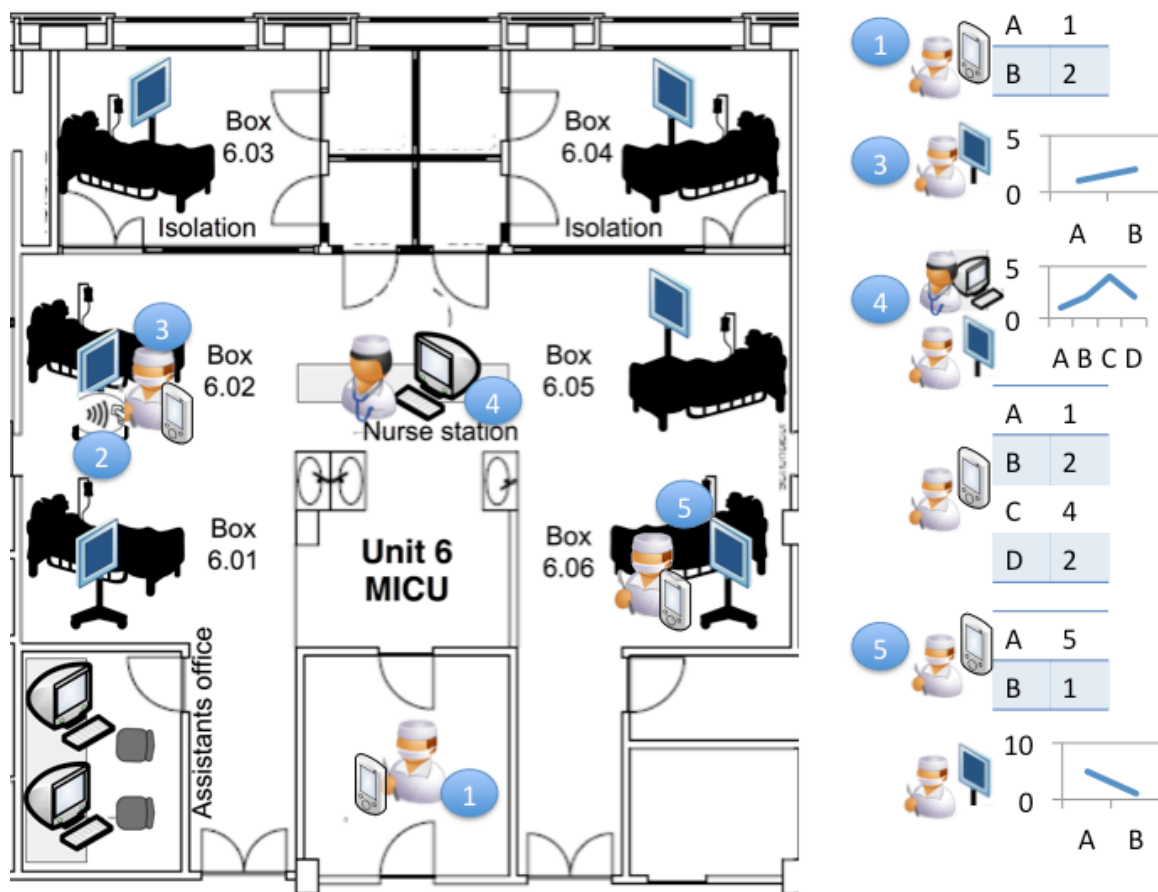


FIGURE 2 - AN ILLUSTRATIVE SCENARIO TO SHOW THE PLATFORM: ON THE LEFT SIDE OF THE FIGURE A PART OF THE ICU FLOOR PLAN IS SHOWN, ON THE RIGHT SIDE AN EXAMPLE OF THE VISUALISATION ON EACH OF THE DEVICES, USED IN THE SCENARIO IS DISPLAYED. 1) THE DOCTOR IS ON HIS WAY TO THE ICU WARD AND ALREADY GOES THROUGH THE PATIENT'S DATA ON HIS SMARTPHONE, BY MEANS OF A TABLE. 2) WHEN HE ARRIVES AT THE PATIENT'S BEDSIDE PC, HE USES HIS PERSONAL TAG TO IDENTIFY HIMSELF TO THE COMPUTER. 3) ON THIS SCREEN, HE SEES A MORE DETAILED OVERVIEW OF THE DATA SHAPED AS A GRAPH. 4) WHEN THE NURSE ENTERS NEW DATA OF THE PATIENT INTO THE SYSTEM, THE NEW INFORMATION IS IMMEDIATELY VISUALIZED ON THE SCREEN OF THE PC AND SMARTPHONE OF THE DOCTOR. 5) WHEN VISITING A NEXT PATIENT, THE DOCTOR CHANGES PATIENT'S ON THE APPLICATION AND ALL SCREENS ARE AGAIN UPDATED.

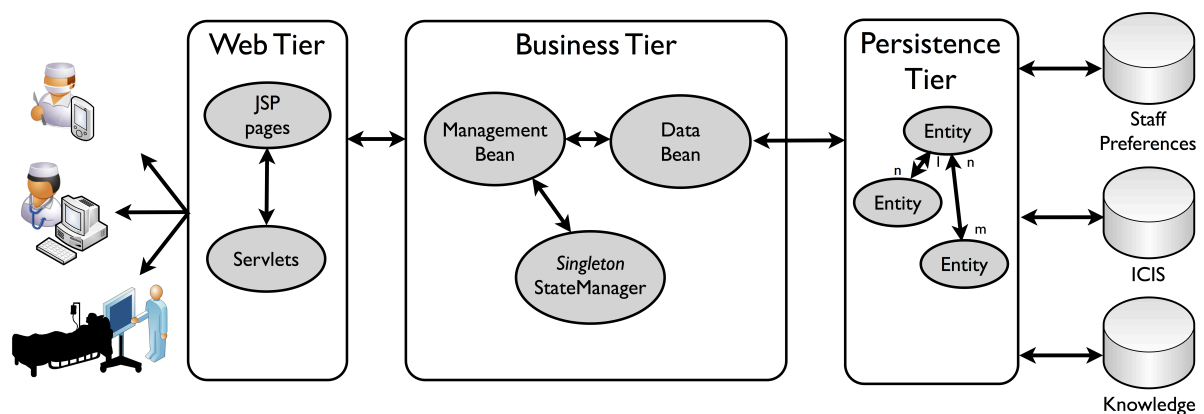


FIGURE 3 - HIGH LEVEL OVERVIEW OF THE PLATFORM'S ARCHITECTURE. THE SERVLETS COMMUNICATE WITH THE BUSINESS TIER AND CONVERT RAW DATA INTO A SUITABLE REPRESENTATION FOR THE DEVICE. THE MANAGEMENTBEAN IS CONNECTED TO TWO INTERNAL COMPONENTS. THE STATEMANAGER KEEPS TRACK OF THE CORRECT RUN-TIME STATE AND THE DATABEAN IS USED TO COMMUNICATE WITH THE PERSISTENCE TIER.

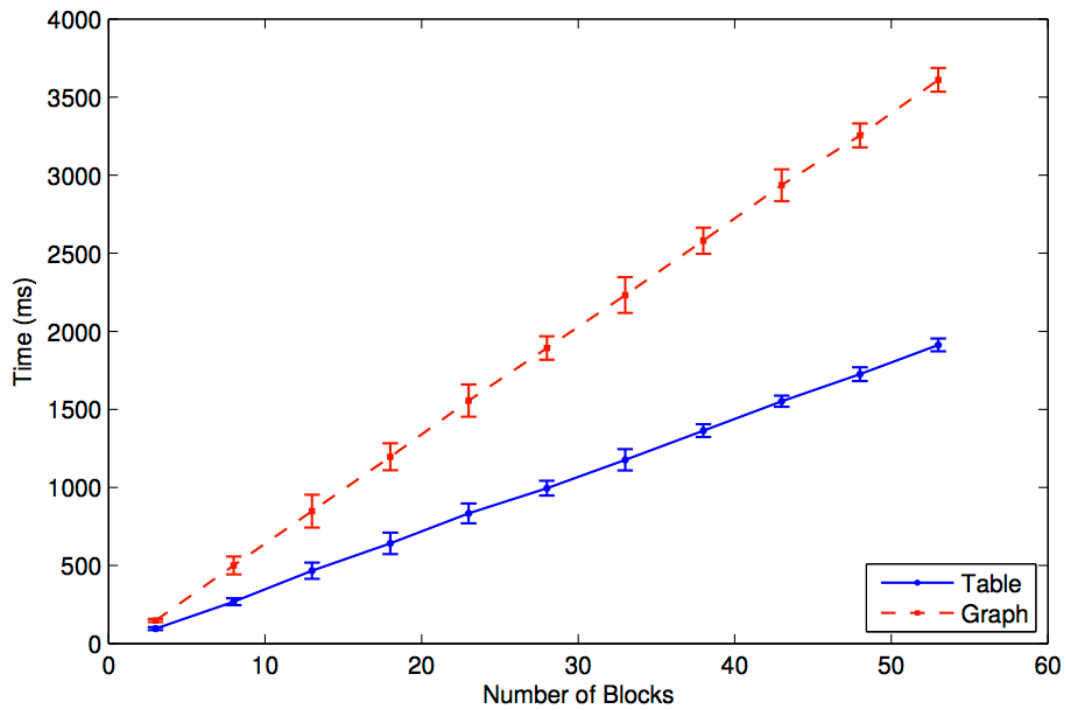


FIGURE 4 - RESPONSE TIME AS A FUNCTION OF THE NUMBER OF BLOCKS WITH THE DATA REPRESENTED AS TABLES OR AS GRAPHS.

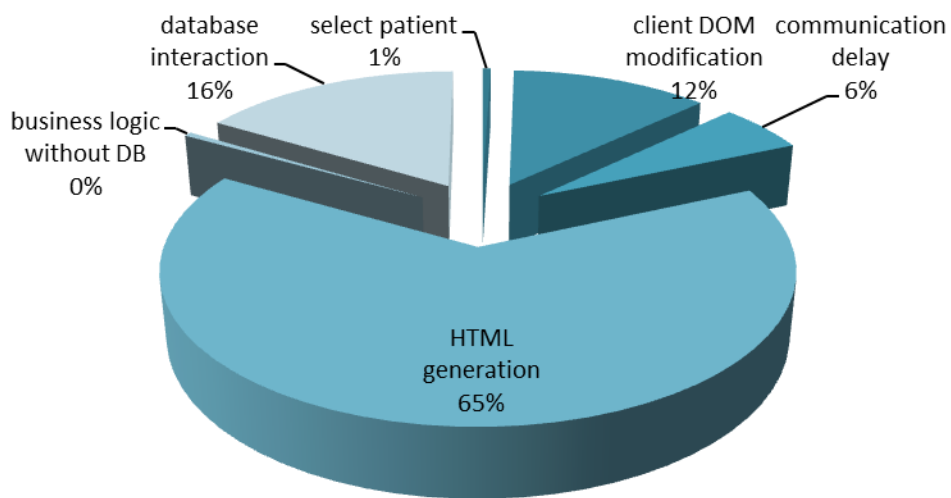


FIGURE 5 - ANALYSIS OF RESPONSE TIME WITH DATA REPRESENTED IN GRAPHS

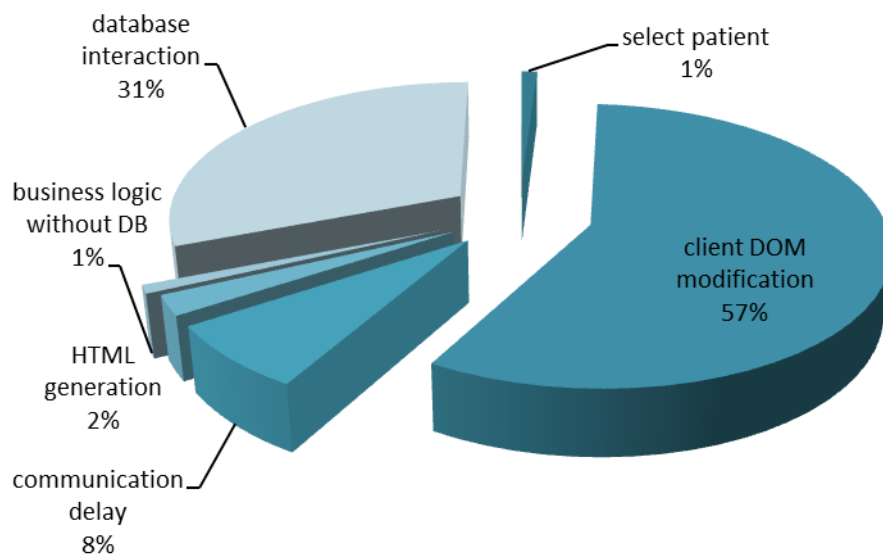


FIGURE 6 - ANALYSIS OF RESPONSE TIME WITH DATA REPRESENTED AS TABLES

TABLES

TABLE 1 - AVERAGE AND STANDARD DEVIATION IN MS FOR THE DIFFERENT PARTS OF THE RESPONSE TIME

	Graph		Table	
	Average [ms]	σ [ms]	Average [ms]	σ [ms]
Select patient	11.07	2.49	11.3	1.62
Client DOM modification	276.4	8.02	645.47	14.78
Communication delay	122.93	17.97	84.2	12.66
HTML generation	1435.7	174.88	26.03	14.30
Business logic without DB communication	10.8	10.08	10.9	12.47
Database interaction	363.07	176.08	348.37	170.41

TABLE 2 - AVERAGE AND STANDARD DEVIATION IN MS CORRESPONDING TO THE RESPONSE TIME FOR THE DIFFERENT TYPES OF DEVICES

	Graph		Table	
	Average [ms]	σ [ms]	Average [ms]	σ [ms]
Computer	1126.27	184.40	2219.97	305.99
Tablet	2798.07	304.80	3652.93	537.28
Smartphone	2449.13	289.17	3218.07	348.94